From Vaporware to Software The Relational Avionics Planning Tool for Operational Requirements (RAPTOR)

Raymond Szymanski
U.S. Air Force Research Lab (AFRL)
Customer Requirements Branch (SNZC)
2241 Avionics Circle, Suite 1
Wright-Patterson A.F.B., Ohio 45433-7303

ABSTRACT

At the 1997 Joint Avionics Weapons Systems conference (JAWS '97) the author presented a paper entitled "Do You Dare Reuse Software Without Extensive Testing." That paper defined the Air Force's new technology planning activity, the Air Force Modernization Planning Process (AFMPP), and the need for information systems that could support that activity. It described one such information system, RAPTOR, that was created by the author through software reuse and discussed the resulting pluses and minuses of a reuse approach. Finally, it laid out plans for future enhancements to both the existing information system and the tools used for application development.

This paper happily reports and assesses the progress that has been made in the RAPTOR development since JAWS '97. It will cover the rational for RAPTOR's development, present details on RAPTOR's main features, discuss traps and pitfalls encountered during the development and how to avoid them, and finally, present a look-ahead to the next generation RAPTOR. One of the major events in the RAPTOR development was to scrap its FoxPro implementation and replace it with a version implemented in Microsoft Access and Visual Basic 5. The rational for replacing this development environment is discussed along with near-term and far-term benefits realized as a result of the change.

1 Who Needs RAPTOR?

1.0 Customer Requirements Branch

The Customer Requirements Branch resides in the Sensors Directorate (SN) of the Air Force Research Laboratory at Wright-Patterson Air Force Base. Its mission is to facilitate the transition of Sensor Directorate technology to the operational user. To accomplish this mission the SNZC engineers must first gain a full understanding of Air Force operational needs, a full understanding of SN-developed technology, and a full understanding of the inter-relationships between the needs and the technologies. This understanding manifests itself in an on-paper linkage between the operational needs and the technology, and establishes a focussed list of potential customers. After the customers have been defined, SNZC needs to bring together the technology developers and the potential customers. This action is necessary to initiate a dialogue between the participants to determine the interest level from both sides and establish formal cooperative agreements. Lastly, SNZC engineers must facilitate the exchange of comprehensive, accurate, and timely information between the technology suppliers and the users, on a regular basis, to ensure a successful technology transition.

All SNZC activity takes place within the framework of the Air Force Modernization Planning Process (AFMPP). In the simplified process discussed above, many of the required communication and information activities and participants in those activities that facilitate the AFMPP are not discussed. However, all in SNZC agree that its main product is information and the manipulation of information into relative formats and useful data for a vast array of AFMPP participants to use and leverage.

1.1 Information Hunters and Gatherers

The information that SNZC collects and distributes is wide and varied in both content and sensitivity. On the collection side SNZC has numerous suppliers of information. The Air Force Mission Area Teams supply mission area operational needs in their annual or biannual Mission Area Plans. Sensor Directorate engineers supply Investment Strategy Data Sheets that document current and future Advanced Technology Demonstrations (ATDs). SN engineers and AFRL associated contractors supply Concept Solutions that address operational deficiencies. SN program managers supply data on their technical programs including white papers and briefing charts.

On the information distribution side SNZC has more participants and customers than on the collection side. Rather than acting as a mere information 'pass-through' SNZC either value adds to the information it collects or filters the information to enhance its usability by the information customer. Recent information-intensive activities that SNZC has supported include the Office of the Secretary of Defense's (OSD) Technology Area Review and Assessment (TARA) and Defense Technology Area Plan (DTAP), the Space Program Objective Memorandum (POM) build, and the Aeronautical Systems Center's (ASC/XR) Concept Call. Unfortunately, each activity required manual integration of information from disparate sources; activities that could have been greatly simplified with the proper support tools.

1.2 Beyond the Big 3

If Andy Rooney, the venerable philosopher of "Sixty Minutes", were to ponder the question "Why were EXCEL, PowerPoint, and Word invented?" he might respond thusly. "So we can keep lots of good information bottled up in big packages and spend lots of time looking for a little fragment in those packages when we need it. At the conclusion of the recent TARA, DTAP, Space POM, and Concept Call activities the participants offices strongly resembled the disheveled look of the Mr. Rooney's own workspace. After tearing through hundreds of files to rack, stack and relate information its no surprise that the offices in question were in need of a quick bulldozing. What would have been useful is a database that contained much of the necessary information with facilities for presenting the information in the required format. That type of help is now available.

In March 1998, the first installation of the RAPTOR database was accomplished in the Customer Requirements Branch. RAPTOR was immediately put to the test as the office manager input Advanced Technology Demonstration (ATD) data. When the tasked was finished the comment made by the user and heard 'round the office was "It was so easy I could have done it with my eyes closed". Grown men wept.

2. RAPTOR

2.0 Overview

RAPTOR Version 1.0 is a database developed in Microsoft Access 97 and Visual Basic 5.0 that addresses the SNZC primary task of " information and the manipulation of information into relative formats and useful data for a vast array of people to use and leverage". RAPTOR runs under Windows 95 and was designed as a single user application that can handle sensitive information. With it the users can collect, connect, view, administer, control, and distribute fine-grained information as it relates to a number of selected 'primary data areas' and 'secondary data areas'.

The primary data areas have strong ties to the AFMPP and include Mission Areas, Needs, Concepts, Subsystems, Tech Needs, Programs, ATDs, Points of Contact (POCs), and NonATDs. The secondary data areas complement the primary data areas but differ in scope since they are document-oriented and provide a big picture view of resident data rather than a granular one. Secondary data areas include Documents and Reports.

Although RAPTOR can perform all the functions listed above, its real strength is its capability to relate or link the various data that it contains. Using this capability, one can create new information that can be stored and viewed within RAPTOR, printed to hard copy, or easily exported to a file for further manipulation and analysis. Now, instead of data in separate files related only by the file name, fine-grained data is explicitly related and exploitable into new information.

2.1 Data Collection

RAPTOR supports two principal means of data collection, one for the RAPTOR administrator and one for the typical user. Figure 2.1 shows the form that is the user's interface for viewing, collecting, managing, and linking data. This form is divided into tabs with one tab for each primary and secondary data area. To move from tab to tab the user merely clicks on the tab of interest. Figure 2.1 illustrates how RAPTOR's main form appears when the Concepts tab is selected and the form is ready to accept a new concept.

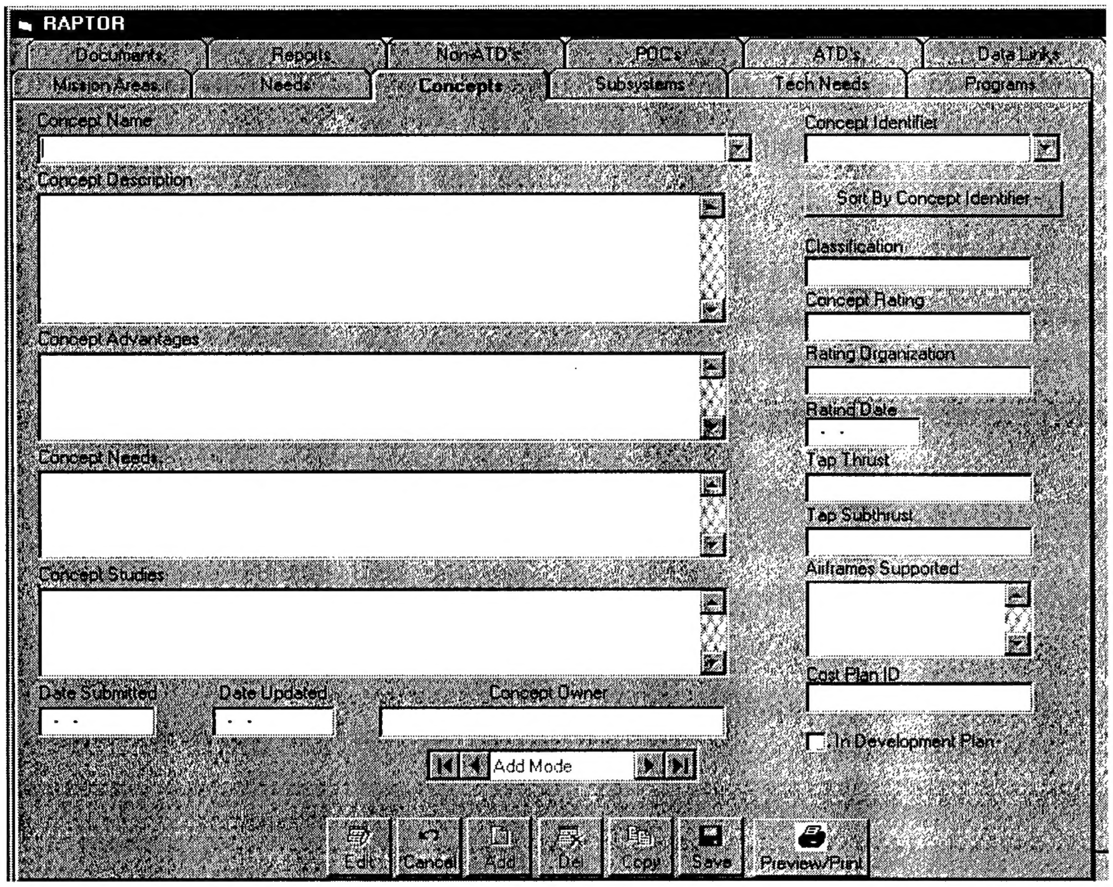


Figure 2.1 RAPTOR Main Form - Concepts Tab

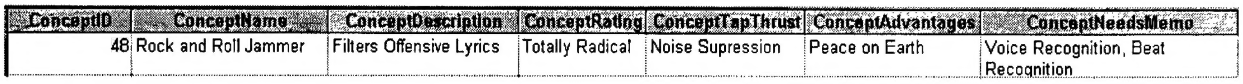
To enter a new record the user must first put the desired tab in the Add Mode. This is accomplished by selecting the Edit button and then the Add button. The form is now ready to accept data into any one of the 18 data boxes seen on the Concepts tab. Data boxes are selected by directly clicking inside of the desired box or by tabbing to it. Once the cursor is in the desired data box the user can type text directly into it or paste text from the clipboard. Data that has been entered is saved to the underlying database by clicking on Save. Selecting Cancel will discard any data that has been entered on the form and will return the tab to Browse Mode. The same procedure is followed for inputting data to any of the primary data area tabs.

Records do not have to be entered one at a time if the desired information is already in electronic columnar format. Since the database administrator has complete access to the underlying database tables, he can append large chunks of data, comprising multiple records, to the tables directly. These large chunks of data do not have to be in Access format as the Access tools will perform format conversion prior to appending new records.

2.2 Data Connections

2.2.1 Within and Between Records

Each data box in Figure 2.1 represents a field in the underlying Concepts data table. Data entered in a data box is stored in the appropriate field. Figure 2.2 shows seven of the nineteen fields from the Concepts table. The first row of Figure 2.2 reveals the actual field names used within the table. The second row contains data that was entered into the database via the Concepts tab. This data is all part of a single record and by definition is related or connected to the other data within that single record.



ConceptID	ConceptName	ConceptDescription	ConceptRating	ConceptTapThrust	ConceptAdvantages	ConceptNeedsMemo
48	Rock and Roll Jammer	Filters Offensive Lyrics	Totally Radical	Noise Supression	Peace on Earth	Voice Recognition, Beat Recognition

Figure 2.2 Concepts Table - Partial Single Record

The capability to relate data is the essential element in the process of making useful information from scattered data. To this end, RAPTOR facilitates data relationships both within and between records. Any record from any of the principal data areas can be explicitly related to any other record in RAPTOR. The capability to relate any record in RAPTOR to any other record provides the user with virtually unlimited ways to exploit the resident data.

Figure 2.3 illustrates the complete set of connections that can be made between RAPTOR data records. If there is a line in Figure 2.3 between a data area and another data area then RAPTOR permits linking of records between those areas. This scheme supports one record-to-one record and one record-to-multiple records relationships.

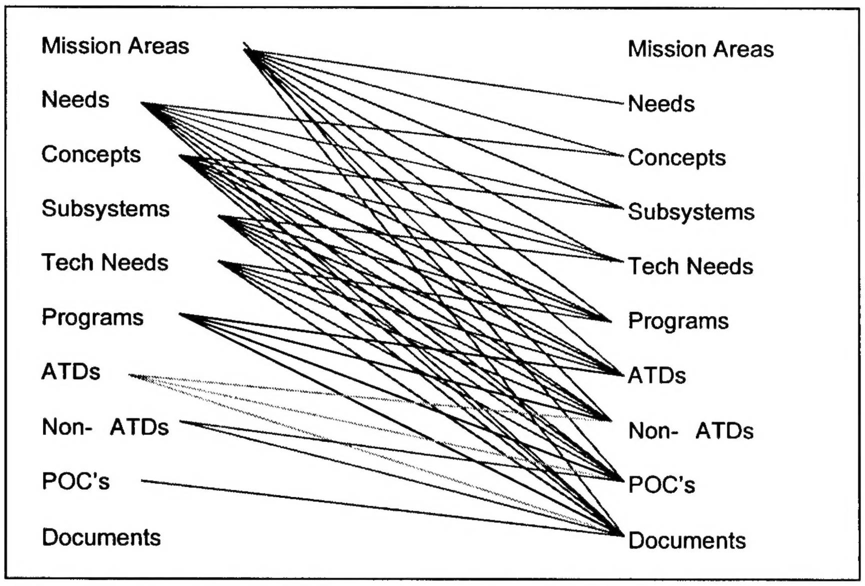
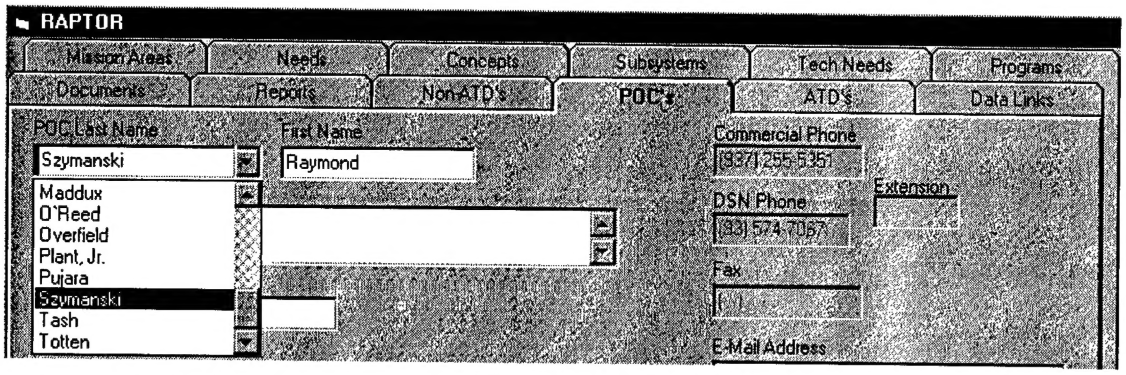


Figure 2.3 Data Area Connection Possibilities

2.2.2 Making Connections

As the following example illustrates, making connections between RAPTOR records is as easy as pointing and clicking. In this fictitious example, the user is the Chief of the Mission Requirements Branch. He wants to assign his employees responsibility for the individual Mission Areas. Because a particular employee was late for the last staff meeting he will be assigned responsibility for five Mission Areas.

To begin the assignment process the user selects the POCs data tab to access the employee's record. An employee is selected by using the POCs Last Name dropdown list box that displays the employees' names. Figure 2.4 shows the upper section of the POCs tab along with the POC Last Name dropdown list. After a selection is made from the list, the selected employee's record is displayed. The Chief may now link this employee record to as many Mission Areas as are available within the database.

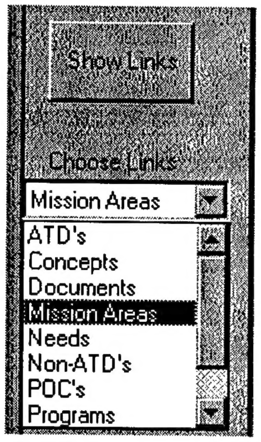


POC Last Name: Szymanski
First Name: Raymond
Commercial Phone: (937) 255-5351
DSN Phone: (93) 574-7067
Fax: ( )

Maddux
O'Reed
Overfield
Plant, Jr.
Pujara
Szymanski
Tash
Totten

Figure 2.4 POC Tab - Last Name Dropdown List

Since the user wants to make links between the selected employee and Mission Areas he chooses 'Mission Areas' from the 'Choose Links' dropdown list and then clicks on the Show Links button (Figure 2.5). These actions cause the 'Mission Links'



Choose Links: Mission Areas

ATD's
Concepts
Documents
Mission Areas
Needs
Non-ATD's
POC's
Programs

Figure 2.5 Choose Links/Show Links

dialogue box to be displayed (Figure 2.6). The dialogue box is used to make the actual links between the selected employee and the desired Mission Areas.

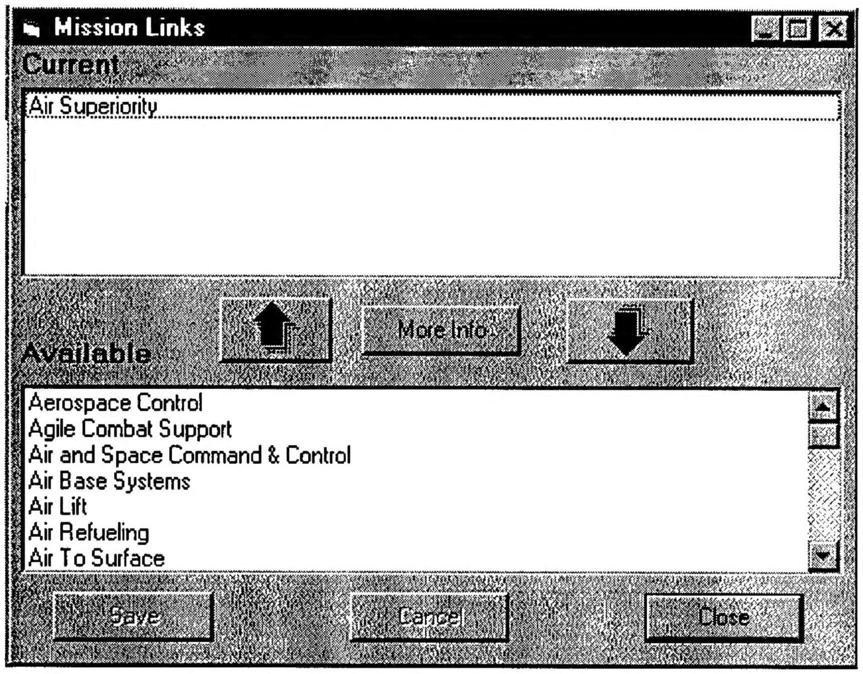


Figure 2.6 Mission Links Dialogue Box

Figure 2.6 shows that the Air Superiority Mission Area is already assigned to the employee by virtue of its appearance in the Current list. The Available list contains the name of every Mission Area in RAPTOR that is not currently assigned to the selected employee. To assign additional Mission Areas, the user must move the desired items from the Available list to the Current list and then save the links by choosing the Save button. Items are moved from one list to the other by either double clicking on the item or by selecting one or more items in a list and then clicking on the appropriate arrow found between the Current and Available lists.

2.3 Data Viewing and Navigation

2.3.1 Single Records

RAPTOR facilitates data record viewing and navigation through multiple data navigation mechanisms and views. The first view the user becomes familiar with is the Tab View that was seen in Figure 2.1 RAPTOR Main Form - Concepts Tab. Recall that this view exposes all the fields for a single record on a single tab. There are two ways to navigate data records in this view, sequentially forwards or backwards and directly.

Figure 2.7 shows the Record Navigation Control that appears on every data area tab. Clicking the right inner arrow will cause the tab to display the next record in that sequence. Clicking the left inner arrow will cause the tab to display the previous record

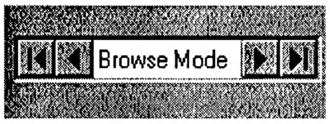


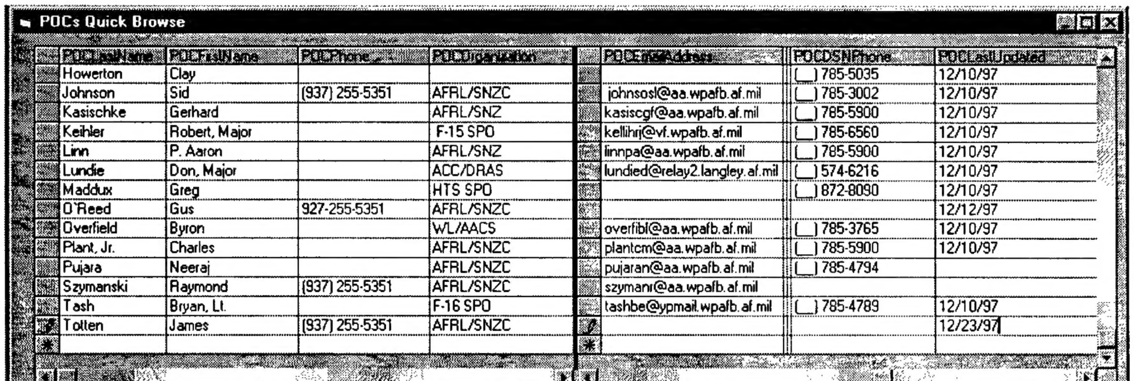
Figure 2.7 Record Navigation Control

in that sequence. Before RAPTOR displays its records in Tab View, it first sorts them in alphabetical order by the Name field. Therefore, the inner arrows select either the previous or next record alphabetically by Name. The left and right outside arrows are used to display the first record or the last record respectively in the alphabetized record set. The Record Navigation Control is the most efficient method of sequential record navigation within RAPTOR.

In section 2.2.2, Making Connections, The POC Last Name dropdown list box was used to directly select the employee's record for assignment to Mission Areas. To facilitate direct selection of records for each data area there is a similar Names dropdown list box. If the user were on the Programs tab, for example, choosing the Program Names list box would produce a list of program names contained in RAPTOR. Choosing from that list causes the display to show the complete record of the program that was selected from the list box.

2.3.2 Multiple Records

Each data area tab is associated with a recordset. This recordset contains every data field for every record for the selected tab. To view a recordset for a particular tab the user must first select the tab and then click the Give Me Data button. Figure 2.8 shows the data grid that is displayed when the Give Me Data button is selected while on the POCs tab. Similar data grids are available for each data area tab in RAPTOR.


POCs Quick Browse

POCLastName	POCFirstName	POCPhone	POCOrganization	POCEmailAddress	POCDSNPhone	POCLastUpdated
Howerton	Clay				(__) 785-5035	12/10/97
Johnson	Sid	(937) 255-5351	AFRL/SNZC	johnsosl@aa.wpafb.af.mil	(__) 785-3002	12/10/97
Kasischke	Gerhard		AFRL/SNZ	kasiscgf@aa.wpafb.af.mil	(__) 785-5900	12/10/97
Keihler	Robert, Major		F-15 SPO	kellihrj@vf.wpafb.af.mil	(__) 785-6560	12/10/97
Linn	P. Aaron		AFRL/SNZ	linnpa@aa.wpafb.af.mil	(__) 785-5900	12/10/97
Lundie	Don, Major		ACC/DRAS	lundied@relay2.langley.af.mil	(__) 574-6216	12/10/97
Maddux	Greg		HTS SPO		(__) 872-8090	12/10/97
O'Reed	Gus	927-255-5351	AFRL/SNZC			12/12/97
Overfield	Byron		WL/AACS	overfibl@aa.wpafb.af.mil	(__) 785-3765	12/10/97
Plant, Jr.	Charles		AFRL/SNZC	plantcm@aa.wpafb.af.mil	(__) 785-5900	12/10/97
Pujara	Neeraj		AFRL/SNZC	pujaran@aa.wpafb.af.mil	(__) 785-4794	
Szymanski	Raymond	(937) 255-5351	AFRL/SNZC	szymanr@aa.wpafb.af.mil		
Tash	Bryan, Lt.		F-16 SPO	tashbe@ypmail.wpafb.af.mil	(__) 785-4789	12/10/97
Totten	James	(937) 255-5351	AFRL/SNZC			12/23/97

Figure 2.8 POCs Data Grid

The data grid provides an effective means of viewing and navigating an entire data area tab recordset. Vertical scroll bars permit sequential navigation through the recordset while horizontal scroll bars permit viewing each field in the wider-than-the-screen record. To further aid in data viewing, the data grid provides the user with user-adjustable field height and width and data table-splitting capabilities.

Figure 2.8 is an example of a data grid that employs the data table-splitting capabilities. Since there are fifteen data fields for each record in the POCs table, their columns cannot be easily displayed at one time. This becomes a problem if the areas of interest happen to be the first few columns of a wide table and the last few columns in the same table. The split table data grid solves this problem by permitting the independent horizontal scrolling of the left and right tables of the data grid. The left table and right table records always stay in synchronization because there is but a single vertical scroll bar to control both table views.

In Figure 2.8 the user is viewing four fields from the leftmost side of the POC table and three fields from the its rightmost side. Upon close examination one can find two additional vertical lines between the columns labeled POCEmailAddress and POCDSNPhone. These lines represent two columns from the POC data table that were horizontally collapsed by the user to bring the Email and DSNPhone columns next to each other for easier viewing.

2.4 Data Reporting and Distribution

2.4.1 Unlinked Data

In the previous sections on data viewing it is evident that what you see is what you entered, basically, data in record format. In data reporting the view has to be more sophisticated and must exploit the data to create information. After all, that's supposed to be the primary reason for using a database.

RAPTOR uses Crystal Reports as its principal report generation and distribution package. The Crystal Reports report generation system that comes bundled with the Visual Basic 5.0 development environment is easy to use and is highly integrated with Visual Basic 5.0. This makes it an ideal reporting system for VB application developers.

RAPTOR provides two principal entry points to the built-in predefined reports. The first is found by clicking on the Print/Preview button. This action produces a dialogue box that offers opportunities to preview and print a number of reports related to the current data area tab. Figure 2.9 shows the tab-oriented print preview dialogue box.

The second report generation entry point is found by selecting the Reports tab. The reports that can be generated from this form are not dependent on a singular tab but can generate reports for any RAPTOR data area. This form also includes the capability to generate reports that reflect the data linking activity discussed in an earlier section. Figure 2.10 shows the Reports tab report-selection form.

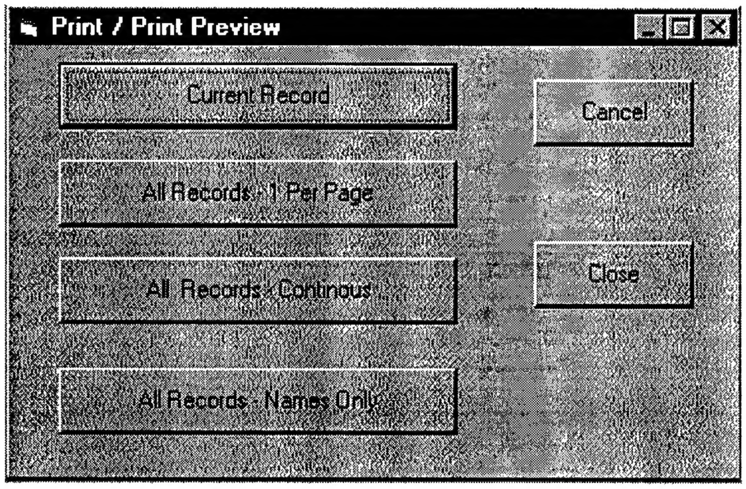


Figure 2.9 Print/Preview Dialogue Box

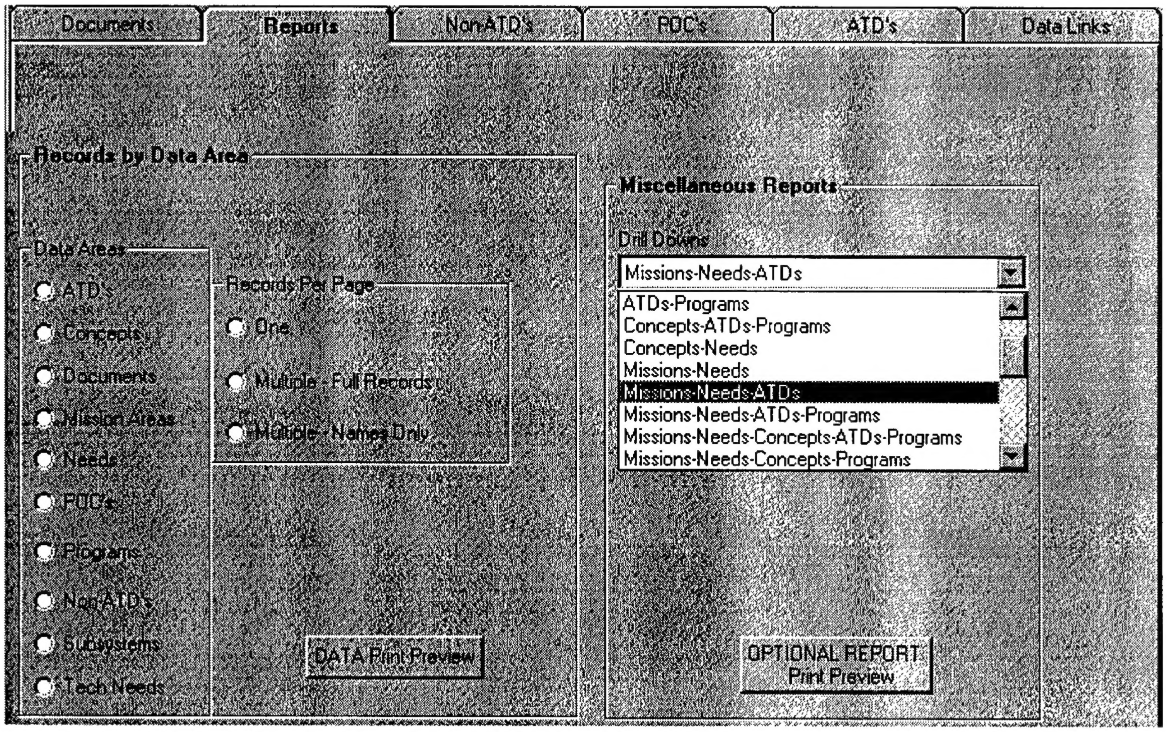


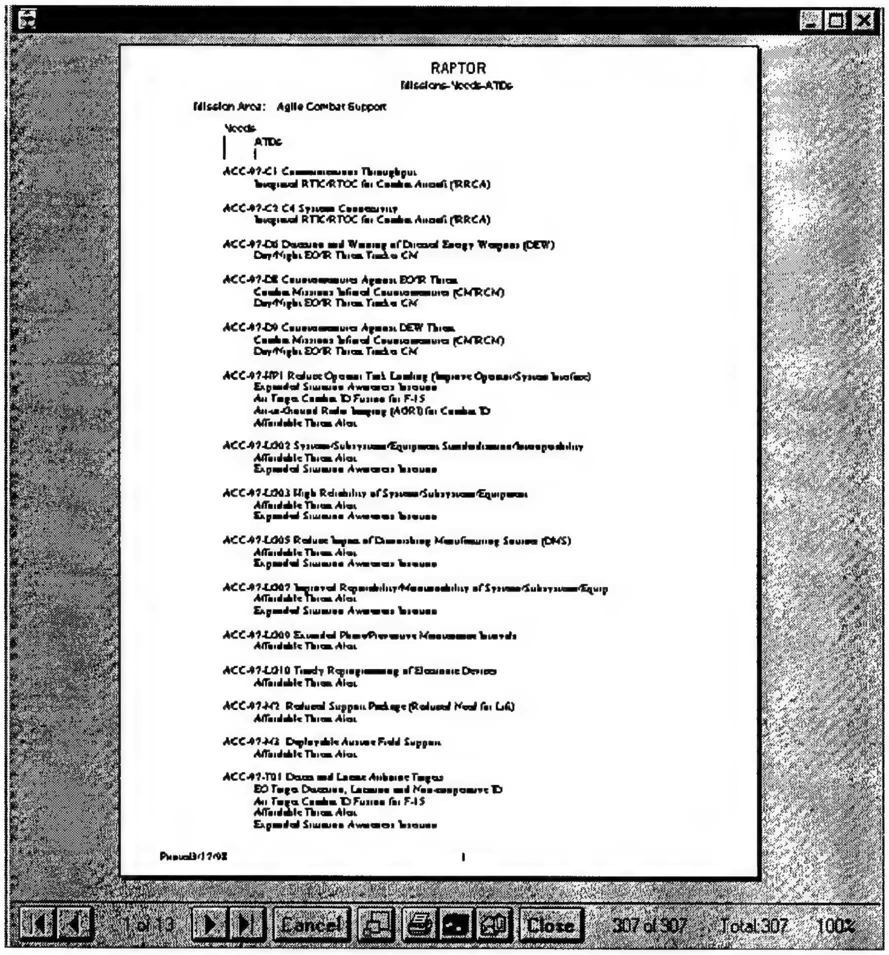
Figure 2.10 Reports Tab

The reports generated by RAPTOR have been designed to meet the requirements of SNZC's diverse customer base. However, it is not always possible to anticipate future customer-information requirements. Fortunately, Crystal Reports provides the capability to export any RAPTOR report to a file. This is an important capability for the user who needs to either customize a report to meet specific customer requirements or to forward the report information in electronic format. If necessary, the user can rely on RAPTOR to provide data linking, initial data sorting and filtering, and then publish the report in whatever format is required.

2.4.2 Linked Data

On the right side of the Figure 2.10 Reports Tab, is a dropdown list box that contains the names of several predefined RAPTOR Drill Down reports. These reports are designed to show the relationships between data areas that were established by the user during the linking process. Since RAPTOR permits any data area record to be linked with any other data area record the number of possible unique reports that can be generated numbers in the hundreds. To keep the reports manageable and keep the report writer from quitting his job, the reports listed in the drill down list box were limited to fulfill current SNZC customer and employee requirements. This list can be readily expanded in the future to include new reports as their requirements are defined.

Figure 2.11 presents the Mission Area-Needs-ATDs drill down report that the user selected from the dropdown list and then generated by selecting the Print button.



RAPTOR
Missions-Needs-ATDs

Mission Area: Agile Combat Support

```
Needs
|    ATDs
|    |
ACC-97-C1 Communications Throughput
     Integrated RTIC/RTOC for Combat Aircraft (IRRCA)

ACC-97-C2 C4 System Connectivity
     Integrated RTIC/RTOC for Combat Aircraft (IRRCA)

ACC-97-D6 Detection and Warning of Directed Energy Weapons (DEW)
     Day/Night EO/IR Threat Tracker CM

ACC-97-D8 Countermeasures Against EO/IR Threat
     Combat Missions Infrared Countermeasures (CMIRCM)
     Day/Night EO/IR Threat Tracker CM

ACC-97-D9 Countermeasures Against DEW Threat
     Combat Missions Infrared Countermeasures (CMIRCM)
     Day/Night EO/IR Threat Tracker CM

ACC-97-HF1 Reduce Operator Task Loading (Improve Operator/System Interface)
     Expanded Situation Awareness Insertion
     Air Target Combat ID Fusion for F-15
     Air-to-Ground Radar Imaging (AGRI) for Combat ID
     Affordable Threat Alert

ACC-97-L002 System/Subsystem/Equipment Standardization/Interoperability
     Affordable Threat Alert
     Expanded Situation Awareness Insertion

ACC-97-L003 High Reliability of System/Subsystem/Equipment
     Affordable Threat Alert
     Expanded Situation Awareness Insertion

ACC-97-L005 Reduce Impact of Diminishing Manufacturing Sources (DMS)
     Affordable Threat Alert
     Expanded Situation Awareness Insertion

ACC-97-L007 Improved Repairability/Maintainability of System/Subsystem/Equip
     Affordable Threat Alert
     Expanded Situation Awareness Insertion

ACC-97-L009 Extended Phase/Preventive Maintenance Intervals
     Affordable Threat Alert

ACC-97-L010 Timely Reprogramming of Electronic Devices
     Affordable Threat Alert

ACC-97-M2 Reduced Support Package (Reduced Need for Lift)
     Affordable Threat Alert

ACC-97-M3 Deployable Austere Field Support
     Affordable Threat Alert

ACC-97-T01 Detect and Locate Airborne Targets
     EO Target Detection, Location and Non-cooperative ID
     Air Target Combat ID Fusion for F-15
     Affordable Threat Alert
     Expanded Situation Awareness Insertion

Printed 3/12/98                         1
```

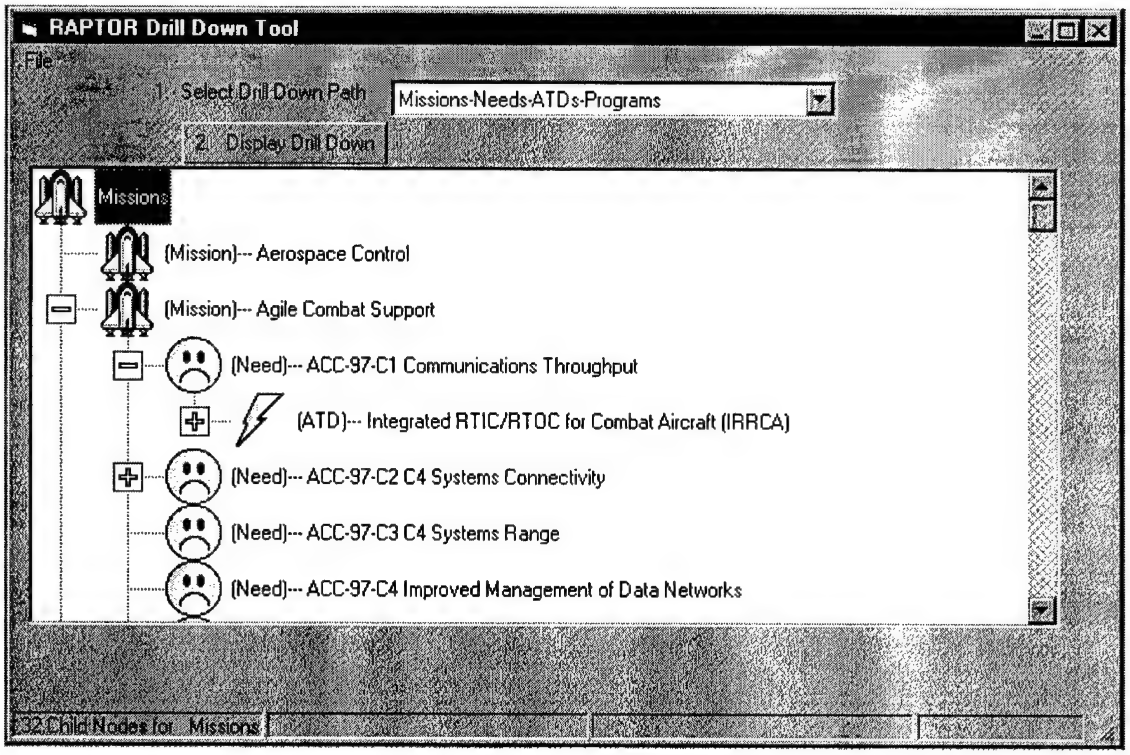
Figure 2.11 Mission Area-Needs-ATDs Drill Down Report Example

This report examines the operational Needs for each Mission Area and reports on those Needs that are linked to a supporting ATD. By design the report automatically page-breaks as the Mission Area changes and indents the Needs and ATDs data to improve readability. All of the built-in drill down reports follow the format displayed in Figure 2.11. However, as revealed in Figure 2.10, the reports vary in the number and type of data areas they report on.

The lower portion of Figure 2.11 shows the navigation and activity buttons that Crystal Reports provides to output the reports and move from report page to report page. Crystal Reports also assists the user by providing some modestly interesting statistical information each time it presents a report preview. In the lower left corner of the Crystal Reports screen it shows the user how many pages are required to print the report and which of those pages is currently displayed. In the lower right corner it reports on the number of unique records that comprise the current report.

2.5 Data Analysis

Although many of RAPTOR's reports can be used to analyze the contents of the database, there is one mechanism that was specifically designed for that purpose, the RAPTOR Drill Down Tool (DDT). While it is very similar to the drill down reports just discussed, even using the same database queries to retrieve its data, it is also very different. The key difference between the drill down reports and the DDT is that the DDT provides the information in a tree view that the user can navigate.



Missions-Needs-ATDs-Programs

Missions
(Mission)--- Aerospace Control
(Mission)--- Agile Combat Support
(Need)--- ACC-97-C1 Communications Throughput
(ATD)--- Integrated RTIC/RTOC for Combat Aircraft (IRRCA)
(Need)--- ACC-97-C2 C4 Systems Connectivity
(Need)--- ACC-97-C3 C4 Systems Range
(Need)--- ACC-97-C4 Improved Management of Data Networks

32 Child Nodes for Missions

Figure 2.12 RAPTOR's Drill Down Tool (DDT)

Figure 2.12 shows the DDT dialogue box that is accessed by selecting the Drill Downs command button.

As an analysis tool the DDT is particularly useful for locating and isolating specific information chains resident in the database. This capability is extremely important to those responsible for populating the database with both information and data links. Remembering once again that database strength is predicated upon storing and relating data, the DDT can help the user quickly ascertain the depth and breadth of information and links contained in RAPTOR. This point will be illustrated in the following simple example.

Having been recently assigned responsibility for four new Mission Areas, the now repentant employee activates the DDT with the Mission Areas-Needs-ATDs-Programs selection. The DDT displays thirty-two space shuttle icons, one for each Mission Area. A quick vertical scroll reveals that one of the employee's newly assigned Mission Areas is not listed, two others do not have any Needs assigned to them, and the final one has assigned Needs but nothing else. This rapid analysis is possible because the icon-oriented DDT provides look-ahead to the next level of detail with a navigable tree structure that is already familiar to any Windows user. The look-ahead capability is provided through the use of '-' and '+' symbols as seen in Figure 2.12. The '-' indicates that a level has already been fully expanded while a '+' indicates that there is at least one additional level of detail available.

In the above example the user could have obtained the same information through the built-in reports or data viewing capabilities but not with the same ease as provided by the DDT. The user quickly realizes he needs to get the missing data into RAPTOR and vows never again to be late for another staff meeting.

3.0 Lessons Learned (Things your mother never told you)

3.1 FoxPro vs. Access/Visual Basic

In 'Do You Dare Reuse Software Without Extensive Testing' the author presented a number of reasons for changing RAPTOR's development environment from FoxPro to a combination of Access and Visual Basic. Foremost among the reasons given was Visual Basic's graphical development environment that "fosters accurate, rapid prototyping and full system development support". Although this statement proved true for the author, several months of development experience with the Visual Basic environment allows one to see that there is much, much more. In fact, comparing VB to Foxpro is like racing a finely tuned Lamborghini against a 1965 Volkswagen bus loaded with hippies; ride the bus and you'll be left in the dust.

Based on extensive experience using both development environments, it is now the author's opinion that the FoxPro environment is obsolete for new Windows-oriented application development. Therefore, there is no merit in providing a feature-by-feature comparison of FoxPro to Access/Visual Basic in this paper. Anyone possessing minimal experience with both environments would certainly render the same opinion. For the skeptical, I recommend your study begin with some light reading in the form of the Foxpro Language Reference Manual that checks in at a meager 1,227 pages.

3.2 Access

It is entirely possible to develop robust desktop databases using Access as the exclusive development environment. This environment is light years ahead of FoxPro in both functionality and user friendliness with its graphical user interface and plethora of Wizards to automate mundane tasks. In fact, RAPTOR would have been developed exclusively in Access had the VB environment not been available with its added features.

After extensive experimentation with both Access and VB and consultation with respected experts, each development tool was applied to tasks that they are best suited to. Access was used to develop RAPTOR's underlying data tables and to develop the database queries that bring data to the user interface. VB was used to create the user interface and to produce the event-driven code.

Creating a table in Access is as easy as typing into a spreadsheet. Linking completed tables together is as easy as point, click and drag. In spite of this, it is also easy to unnecessarily slow the application development and occasionally confound yourself if you do not completely and accurately define the necessary data and link tables first! Before creating any data retrieval queries or any user interface forms, you must be absolutely sure that you are perfectly satisfied with the names that you have assigned to each data field in every table. Why? Because if you change a field name in a table that has been used in a query or form and do not recreate the query or adjust the form with the new name you will experience critical errors at the most inconvenient of times during the application's execution.

Access is not too particular about what you name your tables, what you name the fields in the tables, or what you name the queries. Sure, names are limited to 64 characters, must begin with a letter, can contain numbers, can't contain punctuation marks, and a few other not-too-restrictive rules. This flexibility can be extremely useful or self-defeating depending on whether or not you follow a consistent naming convention for all the items in you application. Highly recommended is the use of the object naming conventions found in the VB Books Online 'Naming Conventions-Coding' help files. Why use VB naming conventions for Access objects? Good question!

The VB naming conventions differ slightly from those enforced by Access but should be used to avoid conflicts when building Access/VB applications. For example, while Access allows the use of spaces in their names, VB does not. VB code limits names to 40 characters while Access permits 64. Code a VB reference to an Access table whose name contains a space or is over 40 characters long and you guessed it, boom! Use the VB naming conventions and the resulting programs and their objects will be easy to read and understand without cramping the programmer's natural creativity. Don't use these and watch programmer productivity drop about 25% as they continually try to guess what type of objects they're working with and where all the strange error messages are coming from.

3.3 Visual Basic

Visual Basic applications begin life as a control object known as a form. Other VB control objects, such data controls text boxes and command buttons are added to the form to facilitate data transfer between the form and the underlying data tables. All of these objects have properties that define the object's physical and state attributes. For example, the Name property provides a unique identifier for each object that can be used to refer to that object in code. Other properties such as Width and Height define a control's two-dimensional display size. Each object also has events associated with it. When an event occurs for an object, like clicking on the object, it responds in a manner defined in that object's event procedure code. Objects also have methods that affect the entire object when invoked. A good example is the Hide method of a form that removes the form from view and reduces it to an entry on the status bar.

The seasoned VB programmer knows that properties, methods, and events vary from object to object although there is some commonality. However, even the veteran programmer can get caught off guard if they assume that two objects having the same properties or methods will react in a similar manner or will acquire the same state. Take the case of a simple text box and a masked edit box. Both controls can display and receive data linked to data tables. When used for data input the masked edit box can also enforce a character-oriented template or filter, restricting the data that it will accept. If the user desires to disable both controls to prevent data input there is a very noticeable resultant surprise on the user's screen. While the text in the text box is clearly displayed, as it was before the disable action, the text in the masked edit box is grayed-out. Surprise!

In many cases these little surprises are not identified in the literature and are popularly referred to as undocumented features. For those situational caveats that are documented they are generally found at the bottom of the help file data sheet for the control object. It is critical to read the entire data sheet for the control along with the respective property, event and method documentation before committing it to a project. After reading the documentation, experiment with the control under near-real project conditions to determine if your understanding of a control's functionality matches reality. Yes, its fun to just jump in and start slapping controls on a form to get your application to do something. However, the time spent fully studying the controls far offsets the time spent ripping out controls that can't quite do what you need them to do when you need them to do it.

Visual Basic applications are event driven. Click a command button and the code in that button's Click_Event will execute. The command button, like all other VB objects, has a name property which the user should set using the naming conventions mentioned earlier. To avoid losing the code and being confronted with one of the most confounding error messages in VB, do not put any code behind a VB object until you are absolutely sure you will not change the name of the object. What happens if this advice is ignored is illustrated below.

A command button named cmdClose will have "Private Sub cmdClose_Click ()" as the first line of its click event procedure. This line is automatically generated by VB. If you add code to this procedure and then rename the button cmdExit, VB will generate another click event procedure opening line, " Private Sub cmdExit_Click ()". Unfortunately, the cmdClose code does not follow the button after you rename it. When you click on the newly renamed command button, cmdExit, it will do nothing and you will be scratching your head at the resultant error message.

Changing the name of an object after putting code in one of its events is bad. Changing the name of an object that has been referenced by another object's code is worse. Executing code containing a reference to an object that no longer exists will return 'Runtime Error 424, Object Required'. The more objects that have been referenced in the offending code, the more fun it is to fix this delightful faux pas.

The number of creative ways to cause trouble by changing the names of objects after they have been referenced by another object is bounded only by one's imagination. Here is one last classic. A Data control links text boxes and other controls to data tables and query-based recordsets. Two Data control properties must be set for this linkage to occur, DatabaseName and RecordSource. To display data from the Data control's recordset two textbox properties must be set, DataSource and DataField. DataSource is set equal to the Name property of the Data control while DataField is set to a field name returned by the Data control. If you change the name of the Data control AFTER setting the DataSource to the Data control's original name, the corresponding text box will look at you like a deer in the headlights. As a matter or fact, if you set any of the above four properties to legal object names and then change the names of those objects, you will get a variety of unexpected inactivity or system error messages.

Quite honestly, there were a large number of other development environment surprises encountered that could be reported here but for the sake of space are not. Of these the most important is the interaction or lack thereof between certain control objects that you want to work together. In one instance two controls, A and B, were chosen to work as a team. Control A would return a recordset while control B would expose the data to the user. Unfortunately, control A could not return the type of recordset that was required for control B to work properly. Once again, to avoid this type of heartache, read all there is to read on a control before betting your application's success on it.

4.0 Future Enhancements

RAPTOR is currently a standalone single-user system. To enable everyone in SNZC to work with their own copy and synchronize data with every other user, an Access facility called replication is used. In replication there is a single master database controlled by the administrator and multiple replicas, one for each user. Changes to the replicas are synchronized with the master, one replica at a time. When all the replicas have been synchronized, new replicas with the cumulative data are distributed. This process will suffice until late fall 98 when SNZC acquires its own server and a client-server version of RAPTOR is deployed.

One of the data area tabs not previously discussed is the Data Links tab. This tab is the future entry point to a number of other local databases that contain information related to data in RAPTOR. The Plans and Programs (P&P) database which tracks programmatic information is certainly of interest to SNZC. Rather than recreating the P&P information in RAPTOR, it will link to the P&P tables over the local area net and extract the data as required. There are other databases that RAPTOR will be cooperating with and similar arrangements for this cooperation are currently under investigation.

The built-in report generation capability is currently limited to the two dozen or so predefined reports. This capability will be significantly enhanced through the use of a third party tool, English Wizard. Reports can be generated based on English-like questions that the user formulates and the tool turns into actual database queries. Although reports can be generated using Access's report generator or VB's Crystal Reports, English Wizard will be easier to use for those who do not care to learn the gory details behind the database's table structure.

As this paper is being prepared, RAPTOR is being installed throughout SNZC. These users are expected to render extensive constructive criticism and suggestions for improvements that will be seriously considered by the developer. The short-term plan is to immediately fix any bugs that are found and redistribute the repaired application. The long-term plan is to incorporate major enhancements into the client-server development. The developer hopes that this approach will keep the dogs from the door and at the same time give the users what the developer wants, the all-important lust for the upgrade.

References

1. Szymanski, Raymond, 'Do You Dare Reuse Software Without Extensive Testing?', 'Joint Avionics, Weapons, and Systems S3 Conference', June 1997.

2. Microsoft Corporation, 'Visual Basic Programmer's Guide', Appendix B - Visual Basic Coding Conventions, 1997.

Biography

Mr. Szymanski is currently the Air Superiority Technical Planning Integrated Product Team representative for the Air Force Research Laboratory, Sensors Directorate, Customer Requirements Branch. He is also responsible for managing the development of the Branch's information technology infrastructure He holds a BSEE from the University of Detroit, did his graduate work in Computer Engineering at Wright State University, and is a certified Windows Application Developer. He is Level III certified in Systems

Planning, Research, Development and Engineering. His technical programs have won two of the prestigious Avionics Technology Transition of the Year Awards. During his tenure as Evaluation & Validation (E&V) Program Manager he chaired the Ada Joint Program Office's E&V Advisory Team, served on the Ada Federal Advisory Board, and was a member of the KAPSE Interface Development Team (Mil-Std-1838A). Mr. Szymanski has lectured at the Naval Postgraduate School in Monterey and the Air Force Computer Resource Acquisition School at Brooks AFB, Texas. He has authored papers for many technical symposiums both domestic and abroad, including the Portable Common Tool Environment (PCTE) conference last held in Paris. His recently completed assignment in the Plans and Programs office as the directorate's technical representative culminated in his receipt of the 1996 Laboratory Excellence Award for Process Improvement. When he is not slinging code or writing papers he enjoys playing golf along with his sons and coaching youth baseball and soccer.

PLEASE CHECK THE APPROPRIATE BLOCK BELOW:

-AQ # ____________

☐ ____________ copies are being forwarded. Indicate whether Statement A, B, C, D, E, F, or X applies.

☒ DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

☐ DISTRIBUTION STATEMENT B:
DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES ONLY; (Indicate Reason and Date). OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT C:
DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND THEIR CONTRACTORS; (Indicate Reason and Date). OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT D:
DISTRIBUTION AUTHORIZED TO DoD AND U.S. DoD CONTRACTORS ONLY; (Indicate Reason and Date). OTHER REQUESTS SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT E:
DISTRIBUTION AUTHORIZED TO DoD COMPONENTS ONLY; (Indicate Reason and Date). OTHER REQUESTS SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT F:
FURTHER DISSEMINATION ONLY AS DIRECTED BY (Indicate Controlling DoD Office and Date) or HIGHER DoD AUTHORITY.

☐ DISTRIBUTION STATEMENT X:
DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND PRIVATE INDIVIDUALS OR ENTERPRISES ELIGIBLE TO OBTAIN EXPORT-CONTROLLED TECHNICAL DATA IN ACCORDANCE WITH DoD DIRECTIVE 5230.25. WITHHOLDING OF UNCLASSIFIED TECHNICAL DATA FROM PUBLIC DISCLOSURE, 6 Nov 1984 (Indicate date of determination). CONTROLLING DoD OFFICE IS (Indicate Controlling DoD Office).

☐ This document was previously forwarded to DTIC on ____________ (date) and the AD number is ________________.

☐ In accordance with provisions of DoD instructions, the document requested is not supplied because:

☐ It will be published at a later date. (Enter approximate date, if known).

☐ Other. (Give Reason)

DoD Directive 5230.24, "Distribution Statements on Technical Documents," 18 Mar 87, contains seven distribution statements, as described briefly above. Technical Documents must be assigned distribution statements.

Tom Minnich
[illegible]
per phone call with J. Camp 10/22/98
Authorized Signature/Date

AFRL/FSD
2241 Avionic Circle
Wright -Patterson AFB, OH 45433-7318
John Camp
Print or Type Name

937 255 2164 3562
Telephone Number